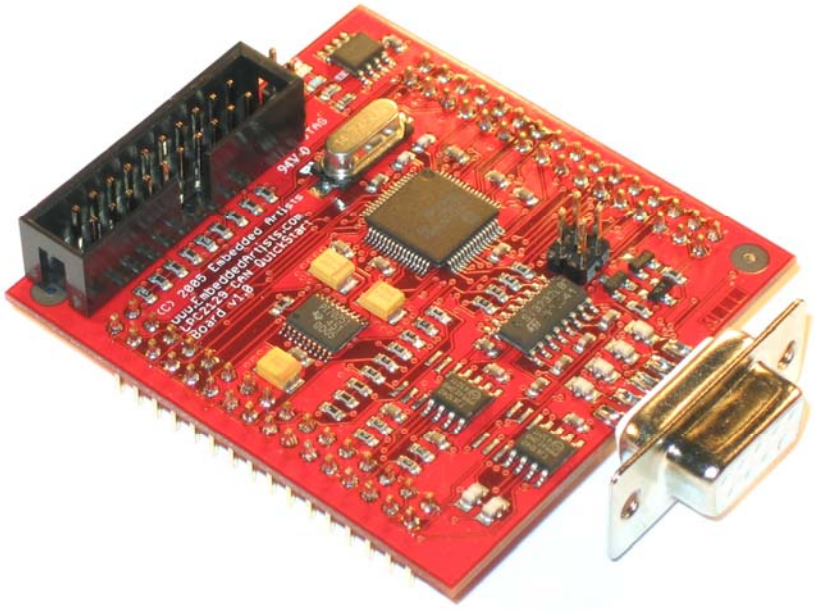
Copyright 2005 © Embedded Artists AB

LPC2129 CAN QuickStart Board User's Guide



Get Up-and-Running Quickly and Start Developing on Day 1…

Embedded Artists

Embedded Artists AB
Friisgatan 33
SE-214 21 Malmö
Sweden

info@EmbeddedArtists.com
http://www.EmbeddedArtists.com

Copyright 2005 © Embedded Artists AB. All rights reserved.

No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual or otherwise, without the prior written permission of Embedded Artists AB.

Disclaimer

Embedded Artists AB makes no representation or warranties with respect to the contents hereof and specifically disclaims any implied warranties or merchantability or fitness for any particular purpose. Information in this publication is subject to change without notice and does not represent a commitment on the part of Embedded Artists AB.

Feedback

We appreciate any feedback you may have for improvements on this document. Please send your comments to support@EmbeddedArtists.com.

Trademarks

InfraBed and ESIC are trademarks of Embedded Artists AB. All other brand and product names mentioned herein are trademarks, services marks, registered trademarks, or registered service marks of their respective owners and should be treated as such.

Table of Contents

1 Introduction 4
1.1 Contents 4
1.2 Features 4
1.3 Low Cost 5
1.3.1 Design and Production Services 5
1.4 Other QuickStart Boards and Kits 5
2 Board Design 6
2.1 Board Schematics 6
2.2 Mechanical Dimensions 8
3 Getting Started 9
3.1 Test program 9
3.2 Program Download 9
3.2.1 Philips LPC2000 Flash Utility 10
3.2.2 LPC21ISP 11
3.3 Program Development 12
3.3.1 QuickStart Build Environment 13
3.3.2 GCC 19
3.4 Installing QuickStart Build Environment 20
4 CD-ROM and Product Registration 25
4.1 CD-ROM 25
4.2 Product Registration 25
5 Further Information 26

Copyright 2005 © Embedded Artists AB

1 Introduction

Thank you for buying Embedded Artists' *LPC2129 CAN QuickStart Board* based on Philips ARM7TDMI LPC2129 microcontroller.

This document is a User's Guide that describes the *LPC2129 CAN QuickStart Board* design along with the accompanying software and program development tools. The document contains information on how to use and integrate the board in your own designs, including electrical and mechanical information.

1.1 Contents

The box received when ordering the *LPC2129 CAN QuickStart Board* contains the following:

- The *LPC2129 CAN QuickStart Board*.
- CD-ROM which includes additional material and programs, including complete and evaluation versions of different development environments. Observe that bulk orders (10 or 100 boards) only include one CD-ROM.

In addition, the following is needed in order to start developing applications with the *LPC2129 CAN QuickStart Board*:

- A DC power supply, 5 volt, capable of providing at least 150 mA (more if external circuits need power from the 3.3 volt supply). Observe that the *LPC2129 CAN QuickStart Board* does not contain any reverse polarity protection. If voltage is applied with wrong polarity, the board will likely be damaged. Also observe that 6.0 volt is the absolute maximum voltage that can be applied without damaging the on-board voltage regulator (TPS70251) and the CAN transceivers (TAJ104x). Consult the TPS70251 datasheet for exact details.
- A serial extension cable, DB9-male to DB9-female (DB9M-DM9F), for connecting the *LPC2129 CAN QuickStart Board* to a PC.
- An optional JTAG interface, for program development debugging.

1.2 Features

Embedded Artists' *LPC2129 CAN QuickStart Board* lets you get up-and-running quickly with Philips ARM7TDMI LPC2129 microcontroller. The small form factor board offers many unique features that ease your development.

- Philips ARM7TDMI LPC2129 microcontroller with 256 Kbyte program Flash, 16 Kbyte SRAM, and 2 CAN communication channels
- All LPC2129 I/O pins are available on connectors
- 12.0000 MHz crystal for maximum execution speed and standard CAN bit rates
  - Phase-locked loop (PLL) multiplies frequency with five; 5 x 12.0000 MHz = 60.0000 MHz
- Dual CAN channels with TJA1040 transceivers
- ESD/EMI protected RS232 channel with DSUB-9 connector
- 256 Kbit I^2C E^2PROM
- Onboard low-dropout voltage and reset generation.
  - Generates +3.3V and +1.8V from a single +5V supply

Copyright 2005 © Embedded Artists AB

  - +3.3V available for external circuits, up to 300 mA
  - Power supply: 5 VDC, at least 150 mA
- Simple and automatic program download (ISP) via RS232 channel
  - Circuit that automatically controls the boot loader from RS232 channel
- Easy to connect to JTAG signals
- Dimensions: 55 x 58 mm
  - Four layer PCB (FR-4 material) for best noise immunity

1.3 Low Cost

The *LPC2129 CAN QuickStart Board* is very low cost and can be used for prototyping / development as well as for OEM production. Modifications for OEM production can easily be done for volumes > 1k. Contact Embedded Artists for further information about design and production services.

Bulk orders (> 10 boards) are delivered without pin rows, DSUB-9 connector, or JTAG connector soldered for easy integration with original equipment.

1.3.1 Design and Production Services

Embedded Artists provide design services for custom designs, either completely new or modification to existing boards. Specific peripherals and/or I/O can easily be added to the different designs, for example communication interfaces, specific analogue or digital I/O, and power supplies. Embedded Artists has a broad, and long, experience in designing industrial electronics in general, and specifically with Philips LPC2xxx microcontroller family.

- Prototype and low-volume production takes place in Sweden for best flexibility and short lead times.
- High-volume production takes place in China for lowest possible cost.

1.4 Other QuickStart Boards and Kits

Visit Embedded Artists' home page, www.EmbeddedArtists.com, for information about other *QuickStart* boards / kits or contact your local distributor.

Copyright 2005 © Embedded Artists AB

2 Board Design

This chapter contains detailed information about the electrical and mechanical design of the *LPC2129 CAN QuickStart Board*. A number of example circuits are also presented that will lower the threshold of start developing with the board.

2.1 Board Schematics

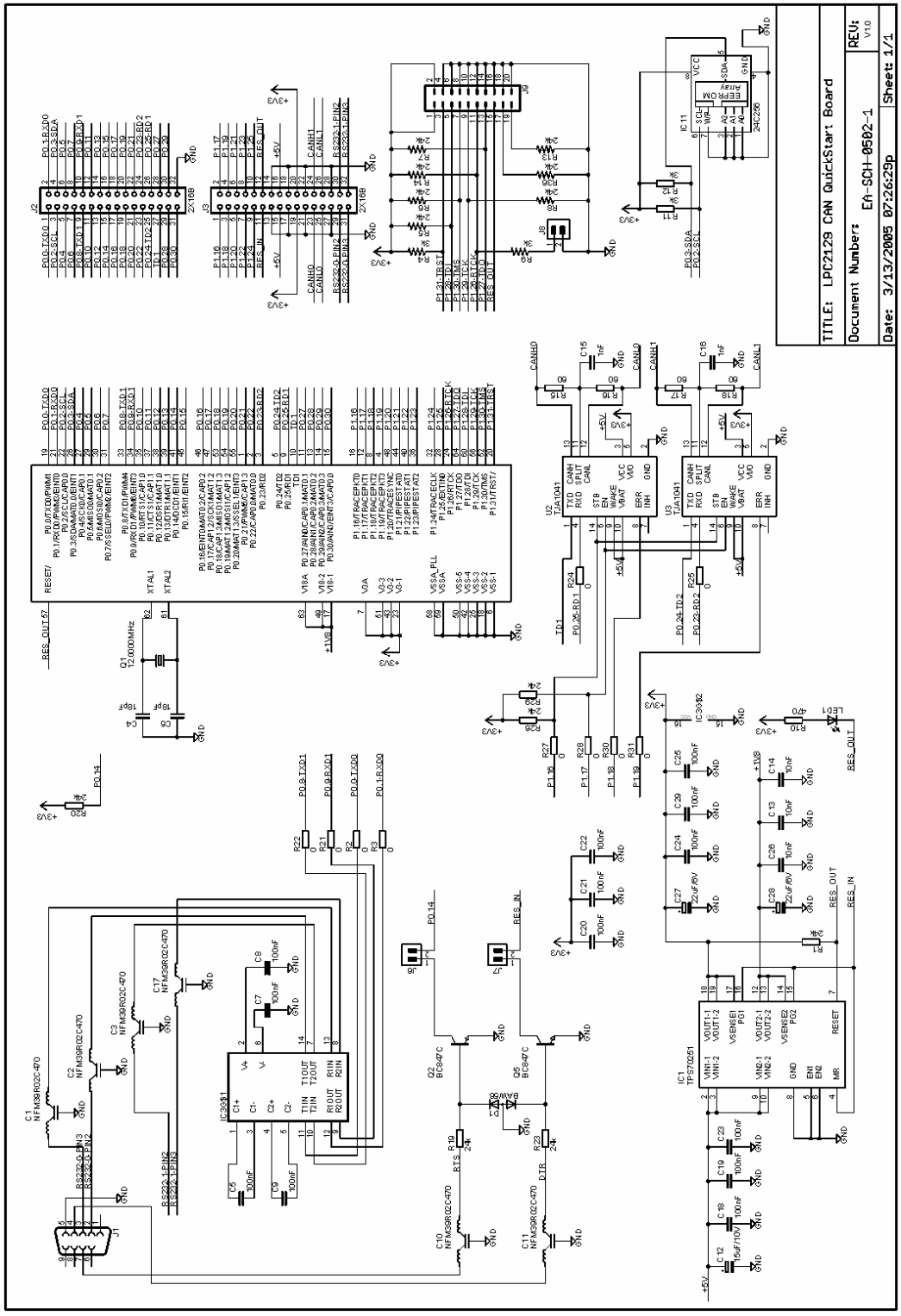


Figure 1 - LPC2129 CAN QuickStart Board Schematic

Copyright 2005 © Embedded Artists AB

Besides the LPC2129 microcontroller from Philips, the board contains dual CAN transceivers, a dual voltage regulator with an internal reset generator, a 256 Kbit I^2C E^2PROM, and an ESD/EMI protected RS232 serial channel. A red LED is connected to the reset signal and lights when reset is active, i.e., the signal is low.

The microcontroller crystal frequency is 12.0000 MHz. This frequency has been selected in order to allow close to maximum execution speed (5 x 12.0000 MHz = 60.0000 MHz, which is the maximum frequency) as well as to provide standard CAN communication bit rates. The crystal frequency can be changed to any desired value for OEM orders, provided that the conditions in the LPC2129 datasheet are met. Current requirements are (but consult the most current datasheet for latest details):

- 1-30 MHz if the on-chip phase-locked loop (PLL) is not used, or
- 10-25 MHz if the PLL is to be used.

The design has direct and automatic support for program downloading (via ISP) over the RS232 serial channel. The RS232 signal DTR controls the reset signal to the LPC2129 microcontroller. The RS232 signal RTS is connected to pin P0.14 in the LPC2129 microcontroller. This pin is sampled after reset and determines if the internal bootloader program shall be started, or not. A low signal after reset enters the bootloader mode. Both the RS232 receive and transmit signals as well as the RTS/DTR signals can be disconnected from the microcontroller via four links / jumpers on the board. See *Figure 2* below for details.

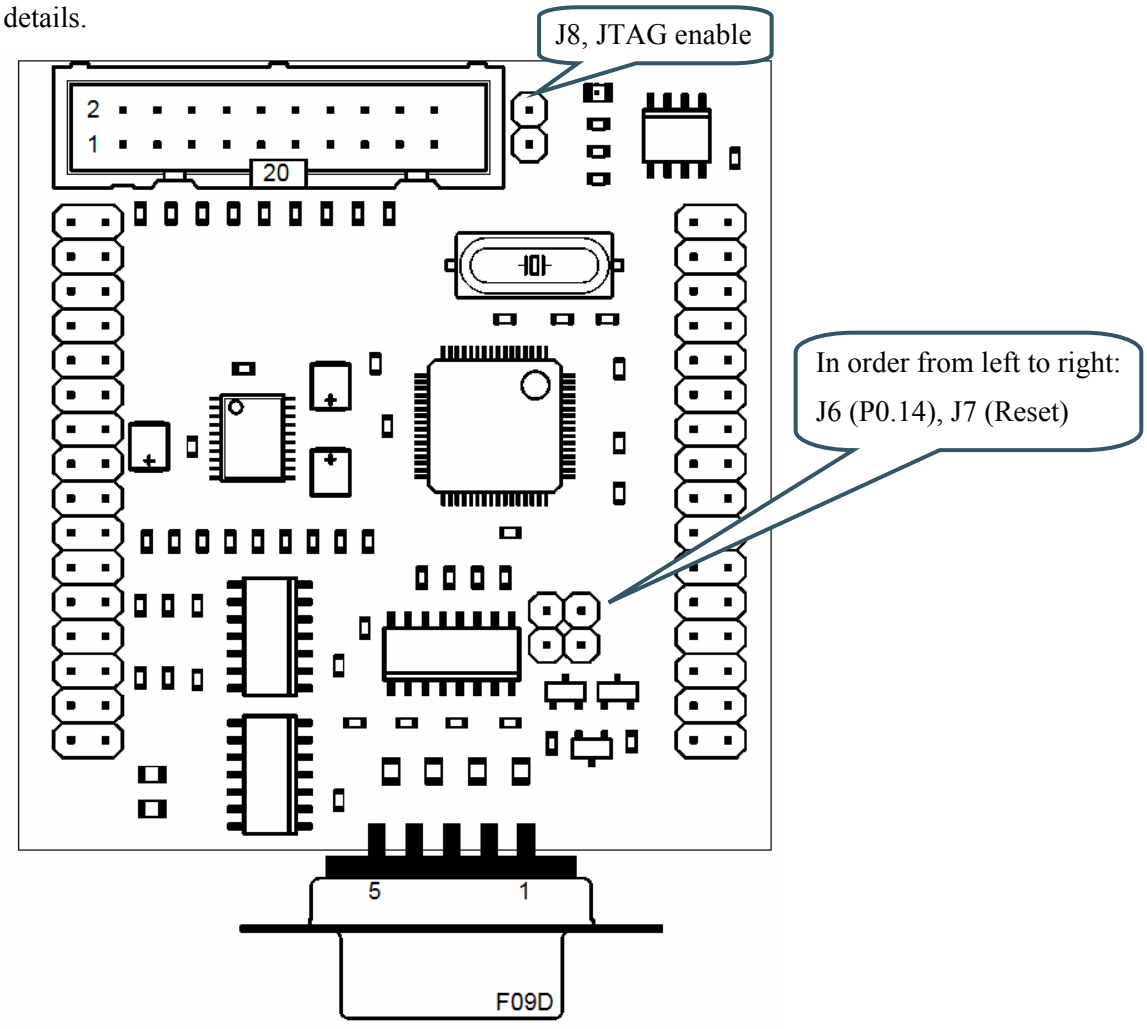


Figure 2 - LPC2129 CAN QuickStart Board Jumpers

Copyright 2005 © Embedded Artists AB

2.2 Mechanical Dimensions

The board interface connectors are placed in two 2x16 pin rows along the board edges. They are 1800 mil apart (measured from outer edges). Figure 3 below contains a drawing of the board that includes mechanical measures.

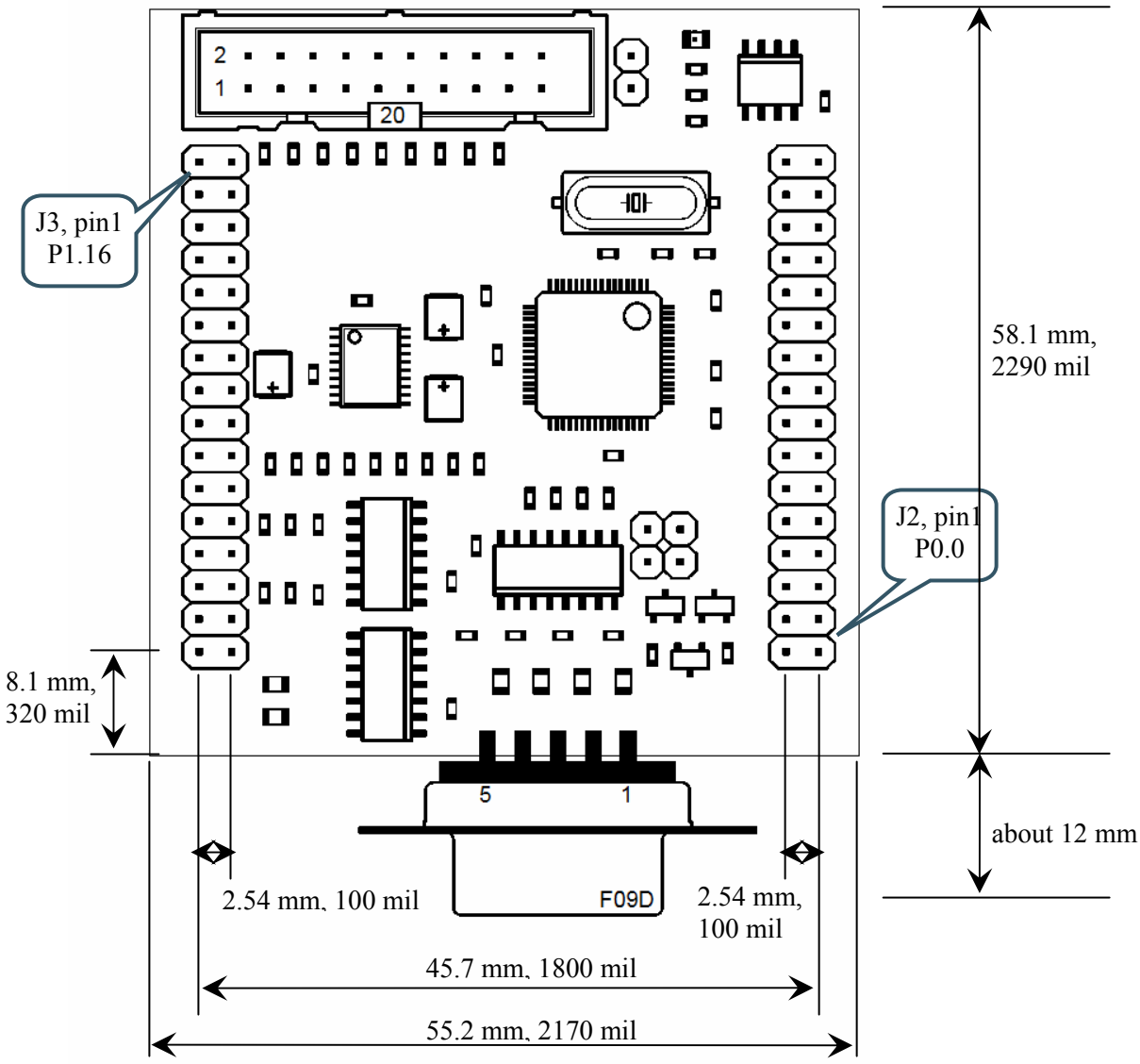


Figure 3 - LPC2129 CAN QuickStart Board Mechanical Dimensions

Copyright 2005 © Embedded Artists AB

3 Getting Started

3.1 Test program

The *LPC2129 CAN QuickStart Board* comes preloaded with a test program. This program can be used to verify that the board operates correctly.

The test program outputs a running-one to port pins P0.4 – P0.31, meaning that one LED at a time will light (in a running-one pattern). Pins P0.0 – P0.1 are tested via the serial channel and pins P0.2 – P0.3 are tested via the I^2C bus. The two CAN channels can be connected together to test the CAN channels. Also, a terminal program should be attached to the RS232 DSUB-9 connector. The test program will output test information regarding the I^2C E^2PROM test, and CAN test. Also, the UART/RS232 channel can be tested by typing characters in the terminal program.

The settings for the terminal program are: 38.4 kbps, 8 data bits, no parity bits, and one stop bit (i.e., 8N1).

The output from the test program will look something like in *Figure 4* below.

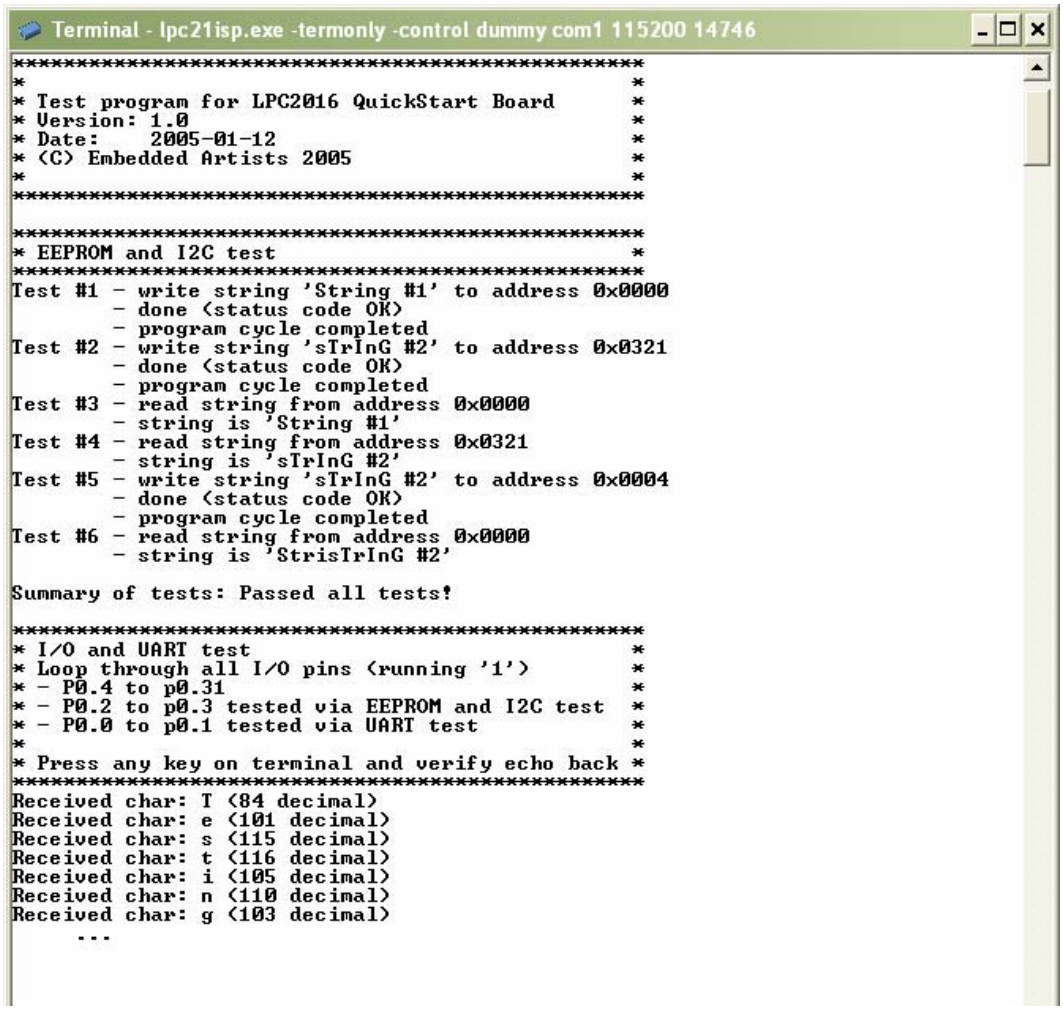

Terminal - lpc21isp.exe -termonly -control dummy com1 115200 14746
****
* Test program for LPC2016 QuickStart Board *
* Version: 1.0 *
* Date: 2005-01-12 *
* (C) Embedded Artists 2005 *
****
* EEPROM and I2C test *
Test #1 - write string 'String #1' to address 0x0000
- done (status code OK)
- program cycle completed
Test #2 - write string 'sTrInG #2' to address 0x0321
- done (status code OK)
- program cycle completed
Test #3 - read string from address 0x0000
- string is 'String #1'
Test #4 - read string from address 0x0321
- string is 'sTrInG #2'
Test #5 - write string 'sTrInG #2' to address 0x0004
- done (status code OK)
- program cycle completed
Test #6 - read string from address 0x0000
- string is 'StrisTrInG #2'
Summary of tests: Passed all tests!
****
* I/O and UART test *
* Loop through all I/O pins (running '1') *
* - P0.4 to p0.31 *
* - P0.2 to p0.3 tested via EEPROM and I2C test *
* - P0.0 to p0.1 tested via UART test *
* Press any key on terminal and verify echo back *
Received char: T (84 decimal)
Received char: e (101 decimal)
Received char: s (115 decimal)
Received char: t (116 decimal)
Received char: i (105 decimal)
Received char: n (110 decimal)
Received char: g (103 decimal)
...

Figure 4 – Example Test Program Output

3.2 Program Download

For now, it is assumed that the program to be downloaded is already developed and there exist a HEX-file to be downloaded. This HEX-file represents the binary image of the application program.

Copyright 2005 © Embedded Artists AB

There are basically two ways of downloading a program into the LPC2129 microcontroller:

- ISP – In-System Programming
  The LPC2129 microcontroller provides on-chip bootloader software that allows programming of the internal flash memory over the serial channel. The bootloader is activated by pulling port pin P0.14 low during reset of the microcontroller. The *LPC2129 CAN QuickStart Board* contains circuits for automatically controlling pin P0.14 and the reset signal over the RS232 channel. This allows the program download to be fully automated.
    - Philips provides a utility program for In-System Flash (ISP) programming called *LPC2000 Flash Utility*.
    - Alternatively, there is a program called LPC21ISP that can be used. Source code is available. This program also provides a terminal functionality, which can be very helpful when developing your application program. The same serial channel that is used to download the program is typically also used for displaying/printing information from the running program. The program immediately switch to terminal mode after program download and will hence not miss any characters sent on the serial channel directly after program start.

  The installation files for both programs can be found on the accompanying CD-ROM.
- JTAG
  For specific information about program download (i.e., Flash programming) with a JTAG interface, consult the manual for the specific JTAG interface that is used (e.g., J-link from Segger, Ulink from Keil, or Wiggler from MacRaigor).

Connect jumpers J6 and J7 on the *LPC2129 CAN QuickStart Board*. This will connect the RS232 channel to the LPC2129 microcontroller and active control over pin P0.14 and the reset signal.

After program download, all jumpers / links can be left connected, or removed if needed. If for example the PC end controls the RS232 signals DTR and/or RTS during normal program execution, then it might be required that jumpers / links J6 and J7 are removed after program download.

3.2.1 Philips LPC2000 Flash Utility

Philips LPC2000 Flash Utility program looks like *Figure 5* below.

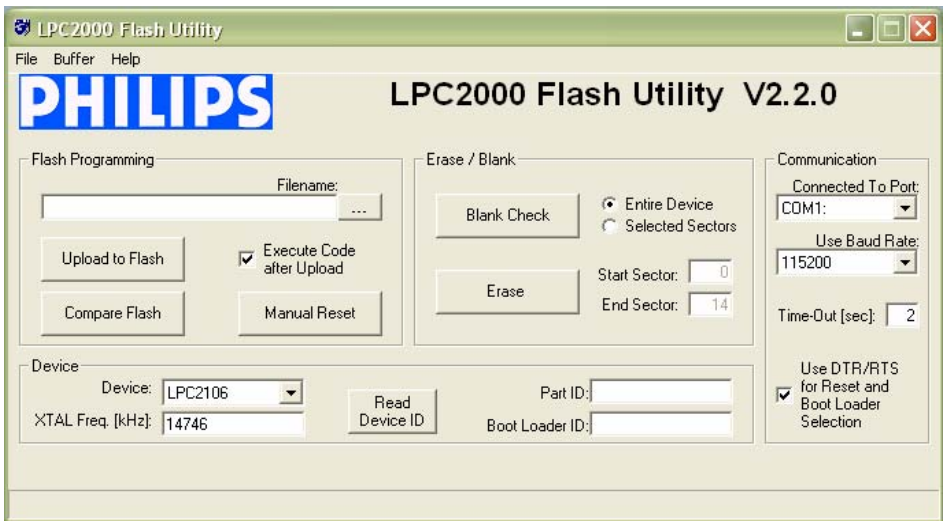


Figure 5 – Philips LPC2000 Flash Utility Screenshot

Copyright 2005 © Embedded Artists AB

Configure the dialog as shown above. The program will control the RS232 signals DTR and RTS if the appropriate checkbox is checked, and hence provide fully automated program download.

Test connection with the *LPC2129 CAN QuickStart Board* by pressing the *Read Device ID* button. The text fields for *Part ID* and *Boot Loader ID* will then contain uploaded information from the microcontroller. Observe that the XTAL Freq. must be set to appropriate value. The default mounted crystal frequency on the *LPC2129 CAN QuickStart Board* is 12.0000 MHz. In this case the value 12000 shall be written in the text box. If the crystal frequency has been changed, make sure the appropriate value is set. If no connection can be established test with a low *Baud Rate*, for example 1200 bps. Also verify that the correct COM-port has been selected (under *Connected to Port*).

Select the HEX file to be downloaded and then press the *Upload to Flash* button.

The downloaded program will immediately start after the download (i.e. the *Upload to Flash* operation is ready) is the option *Execute Code after Upload* is checked.

3.2.2 LPC21ISP

The LPC21ISP program is made publicly available by Martin Maurer. Source code is also available at: http://engelschall.com/~martin/lpc21xx/isp/index.html. *Figure 6* below shows the command syntax for the program.

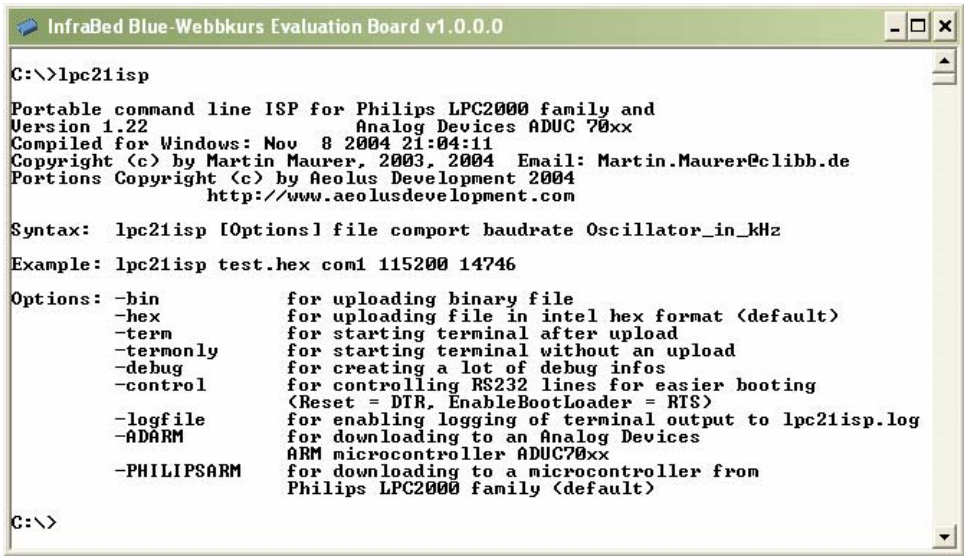

InfraBed Blue-Webbkurs Evaluation Board v1.0.0.0
C:\>lpc21isp
Portable command line ISP for Philips LPC2000 family and
Version 1.22 Analog Devices ADUC 70xx
Compiled for Windows: Nov 8 2004 21:04:11
Copyright (c) by Martin Maurer, 2003, 2004 Email: Martin.Maurer@clibb.de
Portions Copyright (c) by Aeolus Development 2004
http://www.aeolusdevelopment.com
Syntax: lpc21isp [Options] file comport baudrate Oscillator_in_kHz
Example: lpc21isp test.hex com1 115200 14746
Options: -bin for uploading binary file
-hex for uploading file in intel hex format (default)
-term for starting terminal after upload
-termonly for starting terminal without an upload
-debug for creating a lot of debug infos
-control for controlling RS232 lines for easier booting
(Reset = DTR, EnableBootLoader = RTS)
-logfile for enabling logging of terminal output to lpc21isp.log
-ADARM for downloading to an Analog Devices
ARM microcontroller ADUC70xx
-PHILIPSARM for downloading to a microcontroller from
Philips LPC2000 family (default)
C:\>

Figure 6 – LPC21ISP Portable Command Line ISP Screenshot

A typical program download sequence may look like in *Figure 7* below. Here, the test program is downloaded. As seen, the first part is the actual program download phase. Then this is done, the program switches to being a terminal (the second part) and the messages from the test program is displayed. It also sends anything typed on the keyboard back to the *LPC2129 CAN QuickStart Board*. As seen the program ends when ESC is pressed.

This sequence illustrates the benefits from using the program as a terminal directly after program download. No characters are missed after program start.

The used command is:

```
lpc21isp -term -control test.hex com1 115200 14746
```

Copyright 2005 © Embedded Artists AB

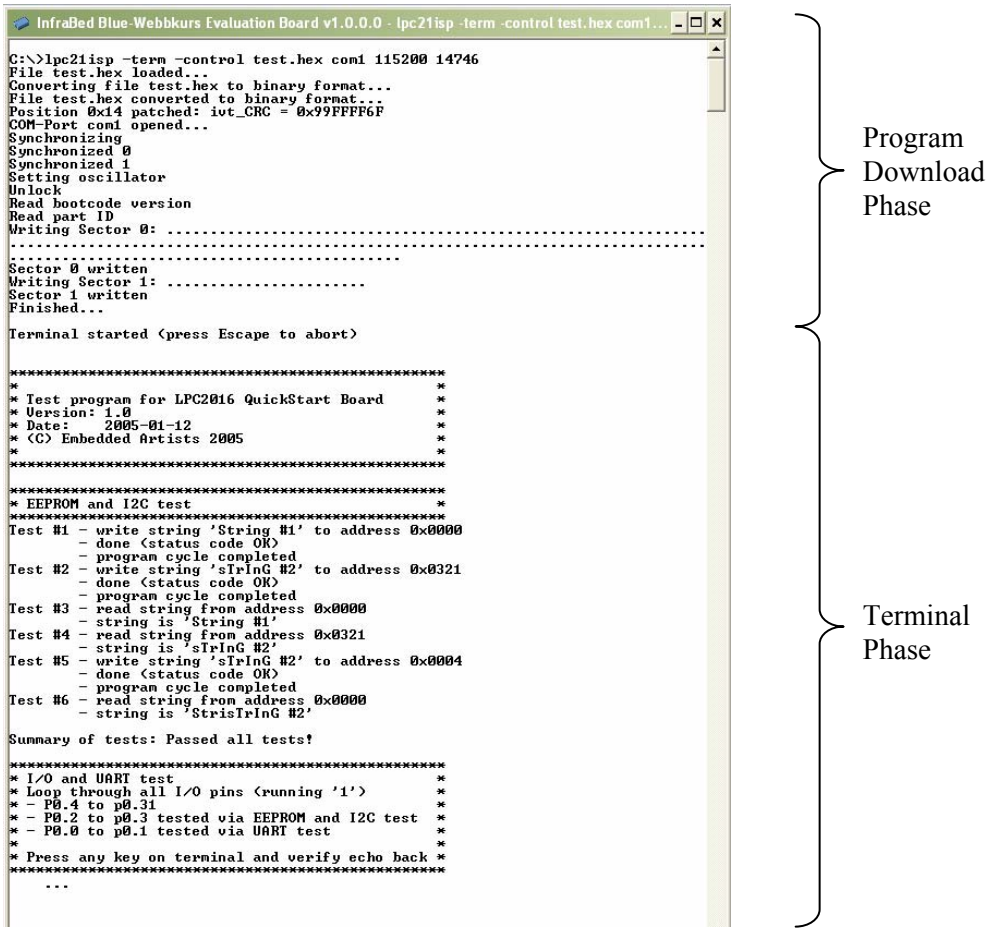

InfraBed Blue-Webbkurs Evaluation Board v1.0.0.0 - lpc21isp -term -control test.hex com1...
C:\>lpc21isp -term -control test.hex com1 115200 14746
File test.hex loaded...
Converting file test.hex to binary format...
File test.hex converted to binary format...
Position 0x14 patched: ivt_CRC = 0x99FFFF6F
COM-Port com1 opened...
Synchronizing
Synchronized 0
Synchronized 1
Setting oscillator
Unlock
Read bootcode version
Read part ID
Writing Sector 0: ...
Sector 0 written
Writing Sector 1: ...
Sector 1 written
Finished...
Terminal started (press Escape to abort)
* Test program for LPC2016 QuickStart Board *
* Version: 1.0 *
* Date: 2005-01-12 *
* (C) Embedded Artists 2005 *
* EEPROM and I2C test *
Test #1 - write string 'String #1' to address 0x0000
- done (status code OK)
- program cycle completed
Test #2 - write string 'sTrInG #2' to address 0x0321
- done (status code OK)
- program cycle completed
Test #3 - read string from address 0x0000
- string is 'String #1'
Test #4 - read string from address 0x0321
- string is 'sTrInG #2'
Test #5 - write string 'sTrInG #2' to address 0x0004
- done (status code OK)
- program cycle completed
Test #6 - read string from address 0x0000
- string is 'StrisTrInG #2'
Summary of tests: Passed all tests!
* I/O and UART test *
* Loop through all I/O pins (running '1') *
* - P0.4 to p0.31 *
* - P0.2 to p0.3 tested via EEPROM and I2C test *
* - P0.0 to p0.1 tested via UART test *
* Press any key on terminal and verify echo back *
...
Program Download Phase
Terminal Phase

Figure 7 – LPC21ISP Portable Command Line ISP Download Screenshot

Another benefit with this program is that it runs under Linux.

Use version 1.28, or later, of LPC21ISP.EXE since older versions must be recompiled with increased reset timeout (when the program tries to synchronize to the *LPC213x QuickStart Board*). The timeout should be increased to at least 350 ms.

3.3 Program Development

There are many options when it comes to the actual application program development. First of all, you must select a development environment, i.e., an editor (preferably with project management capabilities), a compiler package (compiler plus linker), and a debugger. Fortunately, there are many different choices for ARM program development, each with its pros and cons. The list below is far from complete but gives a general overview. The accompanying CD-ROM (see *Section 4.1* for more details) contains many of these programs / environments.

- ***QuickStart Build Environment from Embedded Artists***
  Embedded Artists have created a complete GCC build environment for all QuickStart boards. This will ease program development for novel users. By installing the *QuickStart Build Environment* you will automatically get a complete setup of the build environment.
- ***Rowley Associates CrossWorks for ARM***
  A complete development environment from Rowley Associates, including an editor, project manager, a complete compiler build environment, and a debugger. The version included on the CD is a 30-day fully functional evaluation version.

Copyright 2005 © Embedded Artists AB

- ***IAR Embedded Workbench***
  A complete development environment from IAR Systems, including an editor, project manager, a complete compiler build environment, and a debugger. The version shipped with the *LPC2129 CAN QuickStart Board* has a 32 Kbyte program size limit, but is fully functional in all other aspects.
- ***Keil uVision***
  This is another complete development environment, but from Keil. It includes an editor, project manager, a complete compiler build environment, and a debugger. An evaluation version can be downloaded from Keils homepage. One version of the development environment is based on the GCC compiler (currently version 3.3.1 of GCC).
- ***Programmers notepad***
  This is a very good editor and project manager that is increasing in popularity. The program can easily be integrated with the GCC compiler.
- ***Eclipse + CDT***
  This is a very good development environment (editor and project manager) with specific support for C/C++ code development. It does not contain a compiler but can easily be connected to one, for example GCC.
- ***GCC distribution GNUARM***
  A complete distribution of GCC, specifically for ARM processors. Current version of GCC is 3.4.3 this it is constantly updated.
- ***WinARM***
  This is another distribution that not only contains GCC but also Programmers Notepad, LPC21ISP, a terminal program, and JTAG drivers.

3.3.1 QuickStart Build Environment

The *QuickStart Build Environment* is a complete build environment for GCC including program downloading via ISP. The build environment is built around a bash script. This script sets up all necessary paths. When installing the *QuickStart Build Environment* you will automatically get shortcuts to this bash script. A practical feature is that there can be different scripts for different hardware platforms, for controlling different hardware specific details of the platforms. There can also be many different compilers (including different versions of the same compiler) without conflicting with each other.

The use of the bash script is optional but is recommended for non-experienced users.

A typical project has two subdirectories; **`build_files`** and **`startup`**. *Figure 8* below illustrate the general structure.

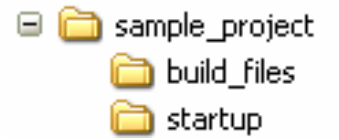


Figure 8 – Typical Project Directory Structure

The subdirectory **`build_files`** contains a general makefile and linker script files. The subdirectory startup contains a configurable startup framework for QuickStart Board projects. The startup files form a library that is linked to the main application.

The makefiles have a hierarchical structure. Each project, either an executable program file or a library, has a simple **`makefile`** that just describe the specifics of the project. This simple **`makefile`** includes the general **`makefile`** that is placed in the **`build_files`** subdirectory.

Copyright 2005 © Embedded Artists AB

Figure 9 below illustrates the simple **makefile**. The example comes from the startup library, found under the **startup** subdirectory. The name of the resulting library is **libea_startup_thumb.a**. Two C-source code files are listed: **consol.c** and **framework.c**. An assembler file called **startup.S** is also included in the library.

```
############################################################
#
# General makefile for building executable programs and
# libraries for Embedded Artists' QuickStart Boards.
# (C) 2001-2005 Embedded Artists AB
#
############################################################

# Name of target (executable program or library)
NAME      = libea_startup_thumb

# ELF-file contains debug information, or not
# (possible values for DEBUG are 0 or 1)
# Extra debug flags can be specified in DBFLAGS
DEBUG   = 1
#DBFLAGS =

# Optimization setting
# (-Os for small code size, -O2 for speed)
OFLAGS  = -Os

# Extra general flags
# For example, compile for ARM / THUMB interworking (EFLAGS = -mthumb-interwork)
EFLAGS  = -mthumb-interwork

# Program code run in ARM or THUMB mode
# Can be [ARM | THUMB]
CODE    = ARM

# List C source files here.
CSRCS   = consol.c \
          framework.c

# List assembler source files here
ASRCS   = startup.S

# List subdirectories to recursively invoke make in
SUBDIRS =

# List additional libraries to link with
LIBS    =

# Add include search paths
INC     = -I .

# Select if an executable program or a library shall be created
#PROGRAM_MK  = true
LIBRARY_MK  = true

############################################################
include ../build_files/general.mk
############################################################################
```

Name of resulting library.

The files are compiled in ARM mode with THUMB interworking (see EFLAGS).

List all included C source files.

List all included assembler files.

Select whether an executable program file or a library shall be created. One of the lines is commented out.

Include the general makefile here.

Figure 9 – Example QuickStart Build Environment Makefile from Startup Library

As seen in *Figure 9* above the makefile ends with the command: **include build_files/general.mk**. This is a general make file that is part of the complete build environment. This part contains all specific details of compiler and linker invocation.

Also at the end, the target must be decided; either an executable program or a library. Either **PROGRAM_MK** or **LIBRARY_MK** must be set to **true**.

The example makefile above is quite simple to its structure. It is possible to create more complex project structures that contains many subprojects. A typical example is to have an application project in a root folder. Under this root folder a number of subdirectories exist

Copyright 2005 © Embedded Artists AB

containing different blocks of functionality. For example, this can be a Real-Time Operating System and a TCP/IP stack. This calls for a recursive **makefile** structure.

The **makefile** in the root filer will create an executable program. It also includes the **makefile** in each of the subdirectories. The **makefiles** that exist in subdirectories will create libraries. An example of a root make file is presented in *Figure 10* below.

```
############################################################
#
# General makefile for building executable programs and
# libraries for Embedded Artists' QuickStart Boards.
# (C) 2001-2005 Embedded Artists AB
#
############################################################

# Name of target (executable program or library)
NAME      = testprogram_lpc2129

# Path and name of linker script file
# Only needed for executable program files
LD_SCRIPT = build_files/link_rom.ld

# ELF-file contains debug information, or not
# (possible values for DEBUG are 0 or 1)
# Extra debug flags can be specified in DBFLAGS
DEBUG   = 1
#DBFLAGS =

# Optimization setting
# (-Os for small code size, -O2 for speed)
OFLAGS  = -Os

# Extra general flags
# For example, compile for ARM / THUMB interworking (EFLAGS = -mthumb-interwork)
EFLAGS  =

# Program code run in ARM or THUMB mode
# Can be [ARM | THUMB]
CODE    = THUMB

# List C source files here.
CSRCS   = main.c

# List assembler source files here
ASRCS   =

# List subdirectories to recursively invoke make in
SUBDIRS = startup \
          can \
          pre_emptive_os

# List additional libraries to link with
LIBS    = startup/libea_startup_thumb.a \
          can/can.a \
          pre_emptive_os/pre_emptive_os.a

# Add include search path for startup files, and othe
INC     = -I./startup

# Select if an executable program or a library shall be created
PROGRAM_MK  = true
#LIBRARY_MK  = true

# Output format on hex file (if making a program); can be [srec | ihex]
HEX_FORMAT  = ihex

# Program to download executable program file into microcontroller's FLASH
DOWNLOAD    = lpc21isp.exe

# Configurations for download program
DL_COMPORT  = com1
DL_BAUDRATE = 115200
DL_CRYSTAL  = 14746
```

Name of resulting program file.

Define linker script.

The files are compiled in THUMB mode.

The root folder only contains one file, the main-file.

Three different subdirectories that contains different blocks of functions in the final application.

The three libraries that are created in the recursive invocation of make are included in the final application.
Note the startup library.

Copyright 2005 © Embedded Artists AB

```
######################################################################
include build_files/general.mk
######################################################################
```

Figure 10 – Example Root Makefile and Recursive Invocation

To build the application program, start a command prompt (the bash script), change directory to the project root, and type: **make**. Depending on the make file content, either an executable program or a library will be created. To also download the executable program, type: **make deploy** instead of just **make**.

A final note about the make file; **make clean** will erase all object files and **make depend** will recreate dependency files (this is also always done when typing just **make**). Finally, **make terminal** will just start the terminal function in the download program (lpc21isp). The specific settings for using the ISP download program can be set with the **DL_XXX** variables (as seen at the end of *Figure 10* above).

As already mentioned, the startup files form a configurable startup framework. This is often called a Board Support package or BSP for short. It contains the very basic startup and initialization code as well as a console with printf()- and scanf()-like functionality. The BSP is very configurable and can be changed according to your specific needs. Each project can have its specific settings. The configuration file is listed in *Figure 11* below, and can be found in file **config.h** in the **startup** subdirectory.

```
/*****************************************************************************
 *
 * Copyright:
 *    (C) 2000 - 2005 Embedded Artists AB
 *
 * Description:
 *    Framework for ARM7 processor
 *
 *****************************************************************************/
#ifndef _config_h_
#define _config_h_

/*****************************************************************************
 * Defines, macros, and typedefs
 *****************************************************************************/

#define FOSC 14745600                /* External clock input frequency
                                        (must be between 10 MHz and 25 MHz) */

#define USE_PLL 1                    /* 0 = do not use on-chip PLL,
                                        1 = use on-chip PLL) */
#define PLL_MUL 4                    /* PLL multiplication factor (1 to 32) */
#define PLL_DIV 2                    /* PLL division factor (1, 2, 4, or 8) */
#define PBSD    4                    /* Peripheral bus speed divider (1, 2, or 4) */

/* initialize the MAM (Memory Accelerator Module) */
#if (FOSC * PLL_MUL) < 20000000
#define MAM_TIMING   1               /* number of CCLK to read from the FLASH */
#elif (FOSC * PLL_MUL) < 40000000
#define MAM_TIMING   2               /* number of CCLK to read from the FLASH */
#else
#define MAM_TIMING   3               /* number of CCLK to read from the FLASH */
#endif
#define MAM_SETTING  2               /* 0=disabled,
                                        1=partly enabled (enabled for code prefetch,
                                          but not for data),
                                        2=fully enabled */

#define IRQ_HANDLER  1               /* 0 = Jump to common IRQ handler
                                        1 = Load vector directly from VIC, i.e.,
                                            LDR PC,[PC,#-0xFF0] */

/*initialize the exception vector mapping */
#define MAM_MAP      1                  /* 1 = exception vectors are in FLASH
```

Copyright 2005 © Embedded Artists AB

```
                                          at 0x0000 0000,
                                      2 = exception vectors are in SRAM
                                          at 0x4000 0000   */

/*
 * CHIP         SRAM SIZE     SRAM START ADDRESS
 * LPC2104       16 * 1024     0x40000000
 * LPC2105       32 * 1024     0x40000000
 * LPC2106       64 * 1024     0x40000000
 * LPC2114       16 * 1024     0x40000000
 * LPC2119       16 * 1024     0x40000000
 * LPC2124       16 * 1024     0x40000000
 * LPC2129       16 * 1024     0x40000000
 * LPC2194       16 * 1024     0x40000000
 * LPC2131        8 * 1024     0x40000000
 * LPC2132       16 * 1024     0x40000000
 * LPC2134       16 * 1024     0x40000000
 * LPC2136       32 * 1024     0x40000000
 * LPC2138       32 * 1024     0x40000000
 * LPC2210       16 * 1024     0x40000000
 * LPC2214       16 * 1024     0x40000000
 * LPC2220       64 * 1024     0x40000000
 * LPC2290       16 * 1024     0x40000000
 * LPC2292       16 * 1024     0x40000000
 * LPC2294       16 * 1024     0x40000000
 */
#define SRAM_SADDR   0x40000000              /* SRAM starting address */
#define SRAM_SIZE    (16 * 1024)             /* LPC2129 */
#define SRAM_TOP     (SRAM_SADDR+SRAM_SIZE)  /* SRAM end address + 1 */
#define SRAM_EADDR   (SRAM_SADDR+SRAM_SIZE-1) /* SRAM end address */

#define stackSize_SYS    600
#define stackSize_SVC     64
#define stackSize_UND     64
#define stackSize_ABT     64
#define stackSize_IRQ    600
#define stackSize_FIQ     64

#define STK_SIZE      (stackSize_SYS+stackSize_SVC+stackSize_UND+stackSize_ABT+
                       stackSize_IRQ+stackSize_FIQ)
#define STK_SADDR     (SRAM_EADDR+1-STK_SIZE)  /* Stack start address */



#define CONSOL_UART              0
#define CONSOL_BITRATE      115200

#define USE_NEWLIB           0   /* 0 = do not use newlib (= save about 22k FLASH),
                                    1 = use newlib = full implementation of printf(),
                                        scanf(), and malloc() */
#define CONSOLE_API_PRINTF   1   /* 0 = printf() = sendString,
                                    1 = simple, own implementation of printf() */
#define CONSOLE_API_SCANF    0   /* 0 = none,
                                    1 = simple, own implementation of scanf() */

#endif  /* _config_h_ */
```

Figure 11 – Board Support Package (BSP) Configuration File

There are three versions of the consol in order to best fit different situations:

- A very simple version that basically only supports printing strings (without any formatting parts) and printing numbers (decimal or hexadecimal).
- A simple printf() implementation that supports the simplest formatting tags. The implementation has been designed for lest possible stack usage (about 40 bytes).
- A full ANSI printf() implementation from newlib (part of the compiler environment that comes with GNUARM). This routine requires about 600 bytes of stack space and should normally not be used in resource constraint systems.

Copyright 2005 © Embedded Artists AB

The code size for the first two alternatives is minimal (about 2k in program size for the entire framework). When using printf() from newlib, the code size is about 30 k for the entire framework (including a large part of the newlib library).

Just edit the configuration file above and recompile your project. The recursive nature of the makefiles will make sure that the startup library is recompiled and linked with the final executable program.

You can find an example project under the QuickStart Build Environment installation. See Figure 12 below for the path. It is typically:
`c:/program/InfraBed/evboards/LPC2xxx-gcc-newlib-vX_X_X_X`.
The beginning of the path can be specific for your installation and the ending of the path is specific for the version of the build environment. The figure below illustrates version 2_0_0_0.

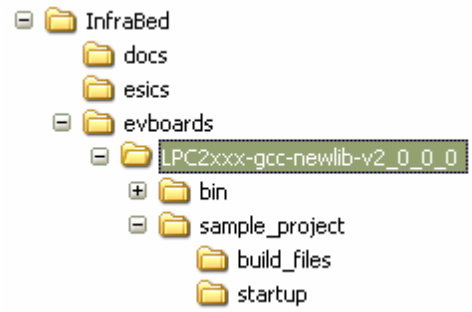


Figure 12 – Sample Project Files under QuickStart Build Environment Installation

The startup framework (BSP) is very simple and can best be understood by studying the source code files. If using the console functionality (printf()- and scanf()-like functions) observe that the function eaInit() must be called before printf() and the console can be used. The following code segment illustrates this.

```
#include <ea_init.h>
...
...
int main(void)
{
  eaInit();  //Now, the console/printf can be used
  ...
}
```

Also observe that whenever the BSP printf() should be used, the following include file must be included into the source code file.

```
#include <printf_P.h>
```

As a summary; Embedded Artists' *QuickStart Build Environment* is comprised of:

- A make build environment, controlled by bash script. A program or library build is started via the command: **`make`**.
- A program download feature, by using the LPC21ISP program. A program build and download is started via the command: **`make deploy`**.
- A Board Support Package (BSP) with startup code and console functions (i.e., printf() and scanf()-like functionality).

Copyright 2005 © Embedded Artists AB

3.3.2 GCC

This will be very similar to the InfraBed example, except that you will have to set up all paths manually and create your own startup files. The **`make`** files will also be a bit more complex. An example **`make`** file is presented in *Figure 13* below. More complex examples than the makefile below also exist.

```
#
# Example makefile that creates a program called 'test', containing the
# C-source code files: main.c, eeprom.c, and i2c.c plus the assembler
# file startup.S
#

LIBS    =
DEBUG   = -g
CFLAGS  = -Wall -nostartfiles -mthumb-interwork -mthumb
INCLUDE = -Iinc/ -Iinc/specific/              #specify include paths here
ARMCC   = arm-elf-gcc
OBJS    = main.o eeprom.o i2c.o startup.o
LDFLAGS = -Wl,-Trom.ld                        #this file controls the linker

all: test.hex

test: $(OBJS)
       arm-elf-gcc $(CFLAGS) $(LDFLAGS) $(OBJS) $(LIBS) -o test.elf
%.o: %.c
       arm-elf-gcc -c $(INCLUDE) $(CFLAGS) $<
%.o: %.S
       arm-elf-gcc -c $(INCLUDE) $(CFLAGS) $<
%.o: %.c
       arm-elf-gcc -c $(INCLUDE) $(CFLAGS) $<
%.hex: %
       arm-elf-objcopy -O ihex $<.elf $@
clean:
       rm -f *.o test.elf test.hex
```

Figure 13 – Example GCC Makefile

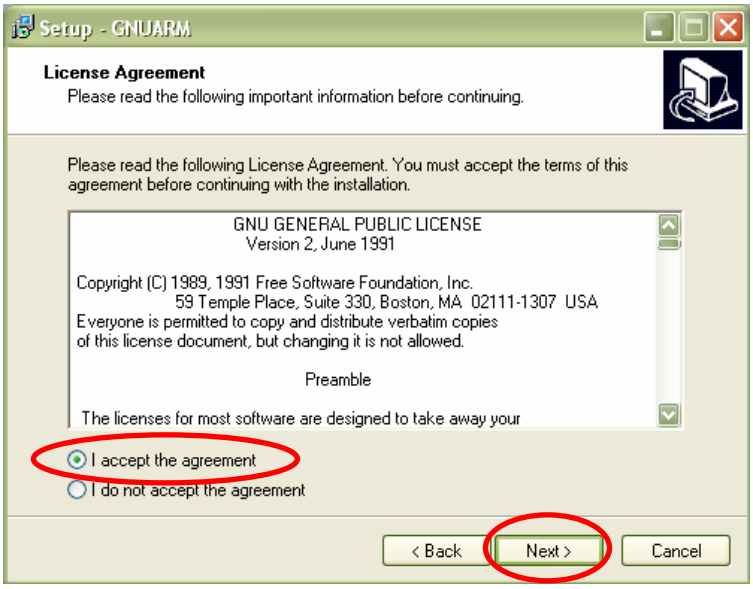
Copyright 2005 © Embedded Artists AB

3.4 Installing QuickStart Build Environment

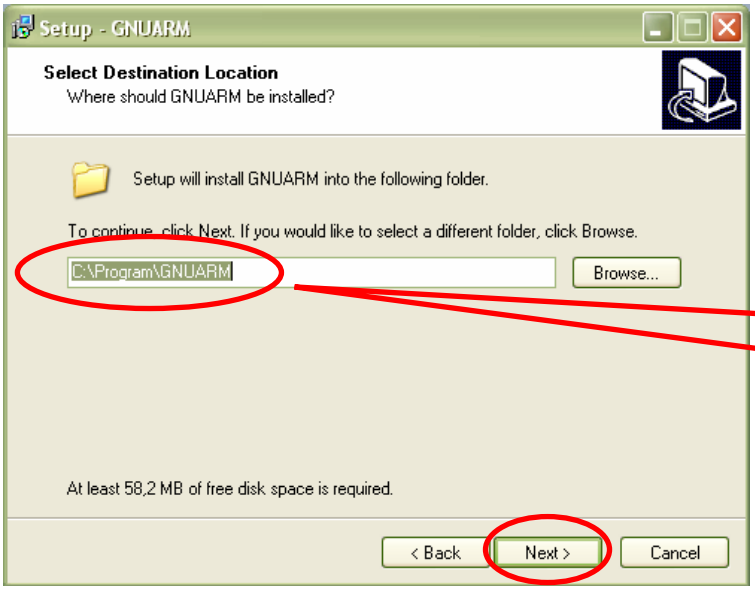
This section describes the necessary steps of program installation that is needed to get the QuickStart Build Environment ready for your use.

- Start with installing the GNUARM distribution that is included in the CD-ROM. The current version of the file is called: **`bu-2.15_gcc-3.4.3-c-c++-java_nl-1.12.0_gi-6.1.exe`**. There is also a newer, but less well tested, version (based on GCC v4.0.0). Only use this newer version if you are an experienced user.
  The installation is very simple and straightforward. It's just following the default installation steps as illustrated in the pictures below:

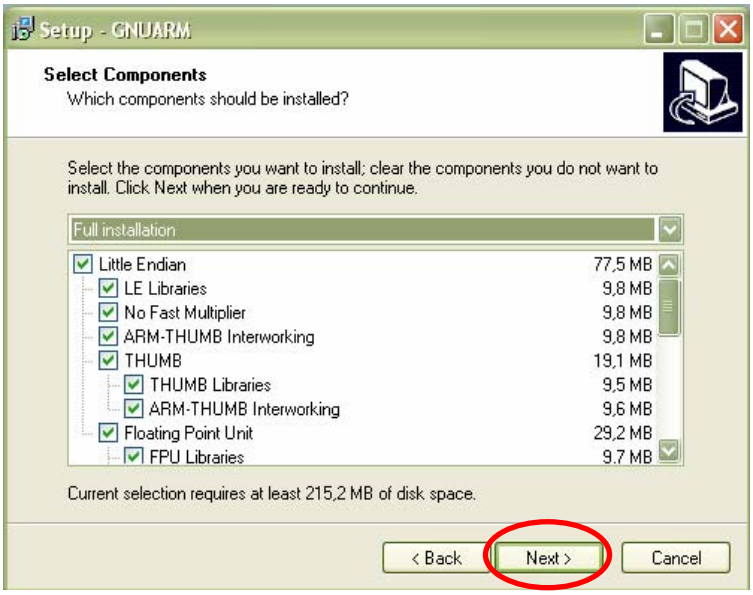
Setup - GNUARM
Welcome to the GNUARM Setup Wizard
This will install GNUARM 3.4.3 on your computer.
It is recommended that you close all other applications before continuing.
Click Next to continue, or Cancel to exit Setup.
Next >
Cancel

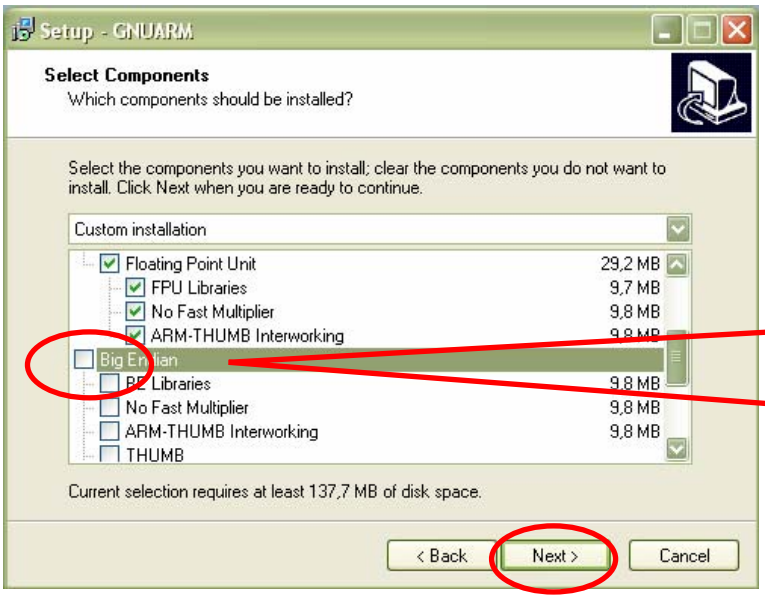

Setup - GNUARM
License Agreement
Please read the following important information before continuing.
Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.
GNU GENERAL PUBLIC LICENSE
Version 2, June 1991
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
Preamble
The licenses for most software are designed to take away your
I accept the agreement
I do not accept the agreement
< Back
Next >
Cancel

Copyright 2005 © Embedded Artists AB


Setup - GNUARM
Select Destination Location
Where should GNUARM be installed?
Setup will install GNUARM into the following folder.
To continue, click Next. If you would like to select a different folder, click Browse.
C:\Program\GNUARM
Browse...
At least 58,2 MB of free disk space is required.
< Back
Next >
Cancel

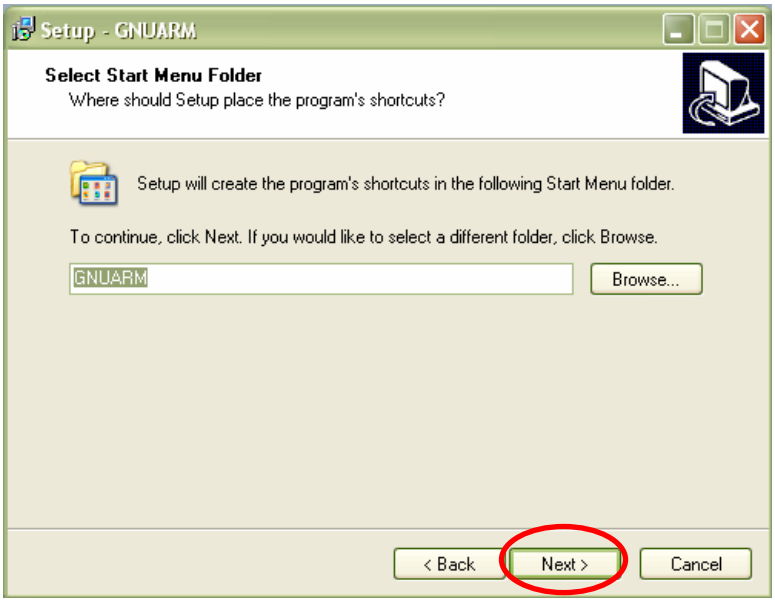
Use the default installation directory

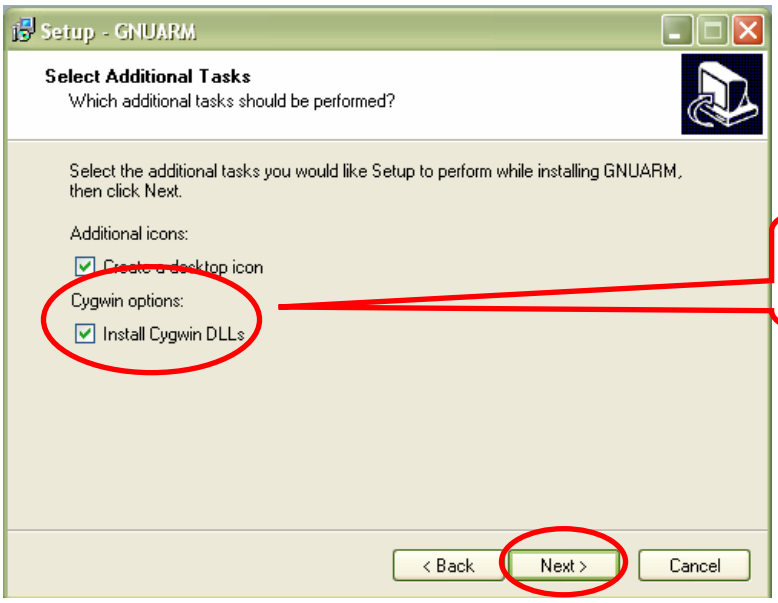

Setup - GNUARM
Select Components
Which components should be installed?
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.
Full installation
Little Endian 77,5 MB
LE Libraries 9,8 MB
No Fast Multiplier 9,8 MB
ARM-THUMB Interworking 9,8 MB
THUMB 19,1 MB
THUMB Libraries 9,5 MB
ARM-THUMB Interworking 9,6 MB
Floating Point Unit 29,2 MB
FPU Libraries 9,7 MB
Current selection requires at least 215,2 MB of disk space.
< Back
Next >
Cancel


Setup - GNUARM
Select Components
Which components should be installed?
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.
Custom installation
Floating Point Unit 29,2 MB
FPU Libraries 9,7 MB
No Fast Multiplier 9,8 MB
ARM-THUMB Interworking 9,8 MB
Big Endian
LE Libraries 9,8 MB
No Fast Multiplier 9,8 MB
ARM-THUMB Interworking 9,8 MB
THUMB
Current selection requires at least 137,7 MB of disk space.
< Back
Next >
Cancel

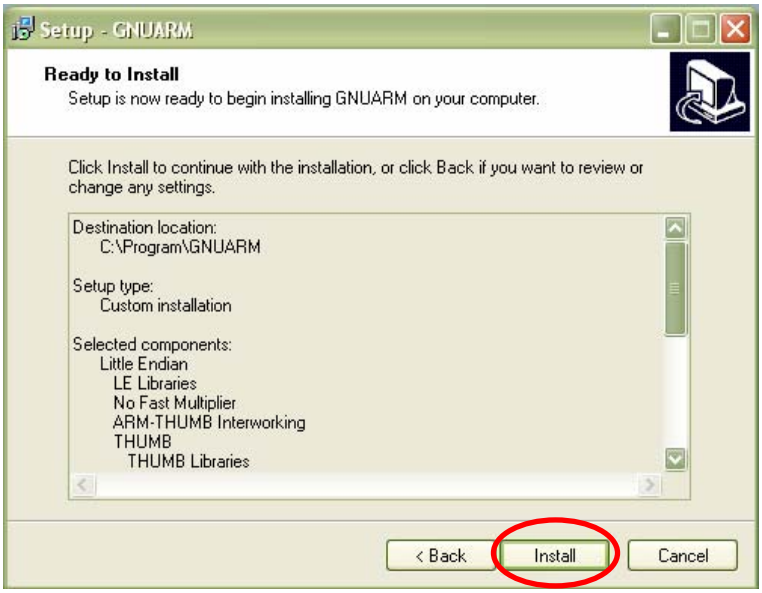
If you want to save space on your harddisk, you can deseclect the *Big Endian* component.

Copyright 2005 © Embedded Artists AB


Setup - GNUARM
Select Start Menu Folder
Where should Setup place the program's shortcuts?
Setup will create the program's shortcuts in the following Start Menu folder.
To continue, click Next. If you would like to select a different folder, click Browse.
GNUARM
Browse...
< Back
Next >
Cancel

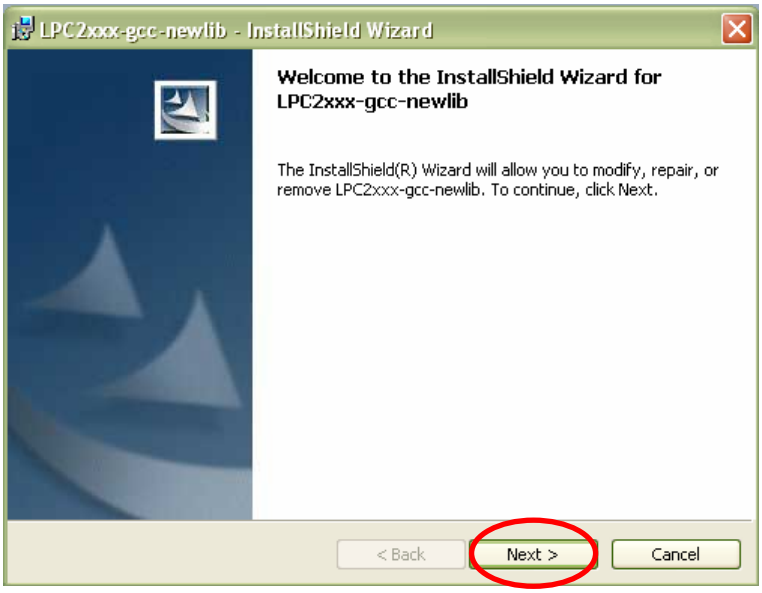

Setup - GNUARM
Select Additional Tasks
Which additional tasks should be performed?
Select the additional tasks you would like Setup to perform while installing GNUARM, then click Next.
Additional icons:
Create a desktop icon
Cygwin options:
Install Cygwin DLLs
< Back
Next >
Cancel

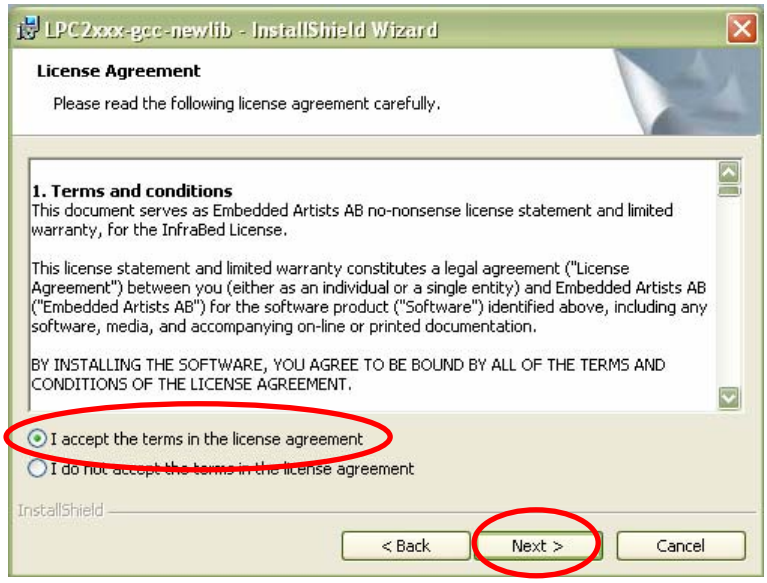
Install the Cygwin DLLs.



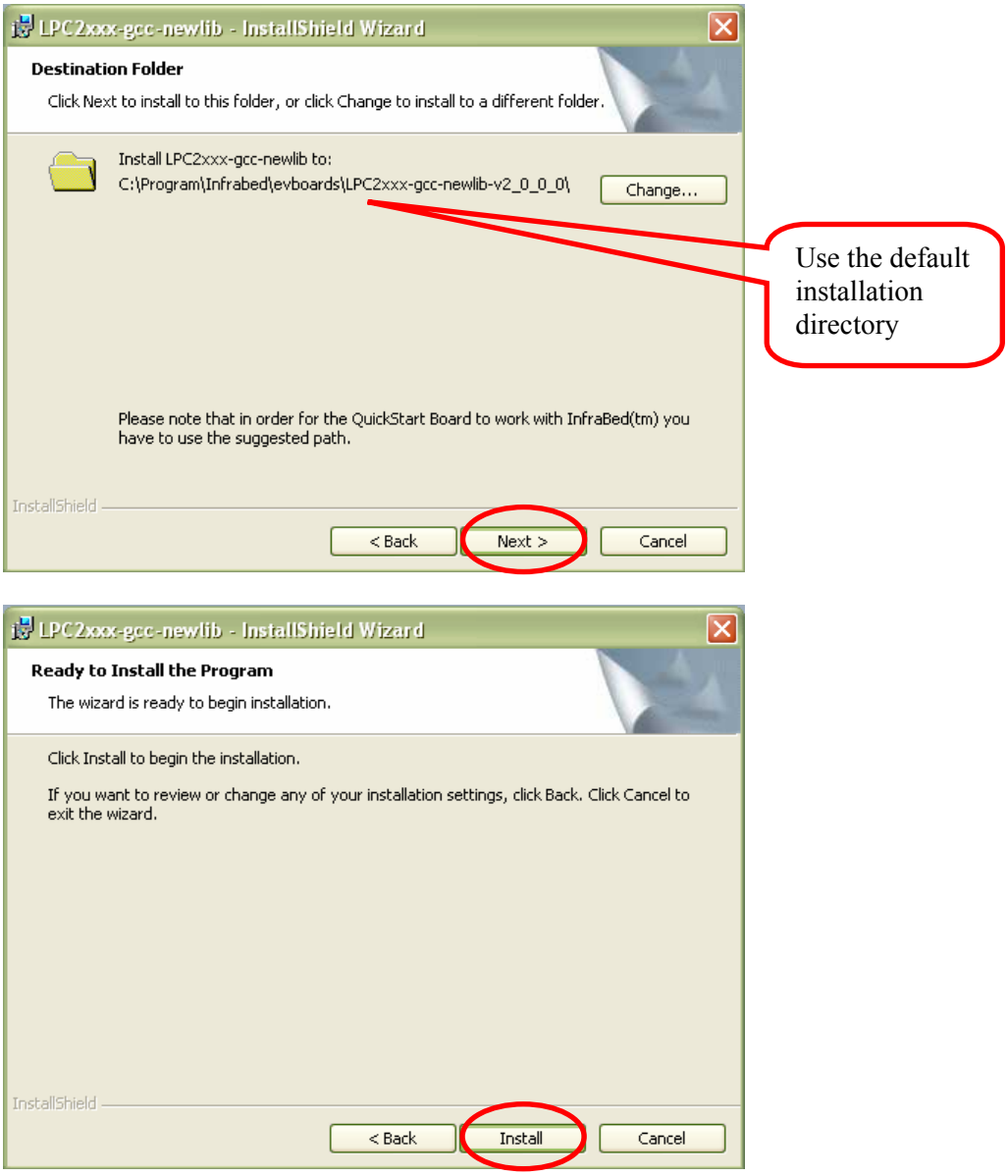
Copyright 2005 © Embedded Artists AB

- Now install the **`LPC2xxx-gcc-newlib_vX_X_X_X`** *QuickStart Build Environment* (vX_X_X_X is the current version of the file). The installation is also in this case very simple and straightforward. Just follow the default installation steps.


LPC2xxx-gcc-newlib - InstallShield Wizard
Welcome to the InstallShield Wizard for LPC2xxx-gcc-newlib
The InstallShield(R) Wizard will allow you to modify, repair, or remove LPC2xxx-gcc-newlib. To continue, click Next.
< Back
Next >
Cancel


LPC2xxx-gcc-newlib - InstallShield Wizard
License Agreement
Please read the following license agreement carefully.
1. Terms and conditions
This document serves as Embedded Artists AB no-nonsense license statement and limited warranty, for the InfraBed License.
This license statement and limited warranty constitutes a legal agreement ("License Agreement") between you (either as an individual or a single entity) and Embedded Artists AB ("Embedded Artists AB") for the software product ("Software") identified above, including any software, media, and accompanying on-line or printed documentation.
BY INSTALLING THE SOFTWARE, YOU AGREE TO BE BOUND BY ALL OF THE TERMS AND CONDITIONS OF THE LICENSE AGREEMENT.
I accept the terms in the license agreement
I do not accept the terms in the license agreement
InstallShield
< Back
Next >
Cancel

Copyright 2005 © Embedded Artists AB


LPC2xxx-gcc-newlib - InstallShield Wizard
Destination Folder
Click Next to install to this folder, or click Change to install to a different folder.
Install LPC2xxx-gcc-newlib to:
C:\Program\Infrabed\evboards\LPC2xxx-gcc-newlib-v2_0_0_0\
Change...
Use the default installation directory
Please note that in order for the QuickStart Board to work with InfraBed(tm) you have to use the suggested path.
InstallShield
< Back
Next >
Cancel
LPC2xxx-gcc-newlib - InstallShield Wizard
Ready to Install the Program
The wizard is ready to begin installation.
Click Install to begin the installation.
If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.
InstallShield
< Back
Install
Cancel

Observe that if the compiler is not installed on the default location (**`c:/Program/GNUARM/`**) the new path must be set in the files **`build.sh`** and **`build_environment.sh`**. Both files can be found in: **`C:\Program\InfraBed\evboards\LPC2xxx-gcc-newlib-vX_X_X_X\bin`**). It is the variable **`COMPILERDIR2`** that must be set (can be found on line 13 in both files). The compiler path must be to the **`GNUARM/bin`** directory.

Observe that the path above must contain the correct version number instead of …**`vX_X_X_X\bin`**. It may for example be: …**`v2_1_0_0\bin`**.

Copyright 2005 © Embedded Artists AB

4 CD-ROM and Product Registration

The accompanying CD-ROM contains a lot of information and programs that will QuickStart your program development! Observe that there may be newer versions of different documents and programs available than the ones on the CD-ROM. See *Section 4.2* for information about the product registration process, which allows you to always have access to the latest versions.

4.1 CD-ROM

The following is included on the CD-ROM:

- The preloaded test program, in source code format and as a HEX-file.
- The two different ISP download programs.
- Datasheets of all circuits on the *LPC2129 CAN QuickStart Board*.
- *QuickStart Build Environment* from Embedded Artists, which contains a complete setup of a build environment for GCC.
- A complete development environment: Rowley Associates CrossWorks for ARM, 30-day evaluation version.
- A complete development environment: IAR Embedded Workbench for ARM, Kickstart Edition with 32 Kbyte program size limit.
- Another complete development environment: GCC, GNUARM distribution, including compiler, linker, make, and debugger.
- The program Programmers Notepad, which is a very good program development editor and project manager.
- The Eclipse development environment including the CDT (C/C++ Development Tools) project.

4.2 Product Registration

By registering as a customer of Embedded Artists you will get access to more valuable material that will get you up-and-running instantly:

- Access to a Real-Time Operating System (RTOS), in the form of a library that can be used for non-commercial applications.
- Access to a number of sample applications that demonstrated different (peripheral) functions in the LPC2129 processor.
- Access to the latest versions of all information and programs on the CD-ROM.

Registering is easy and done quickly.

1) Go to http://www.EmbeddedArtists.com, select *Support* and then *Register*.
2) Type in the products serial number (can be found on the *LPC2129 CAN QuickStart Board* or on the package carrying the board) along with your personal information.

Copyright 2005 © Embedded Artists AB

5 Further Information

The LPC2106 microcontroller is a complex circuit and there exist a number of other documents with a lot more information. The following documents are recommended as a complement to this document.

[1] Philips LPC2129 Datasheet
http://www.semiconductors.philips.com/acrobat/datasheets/LPC2119_2129-03.pdf

[2] Philips LPC2129 User's Manual
http://www.semiconductors.philips.com/acrobat/usermanuals/UM_LPC21XX_LPC22XX_2.pdf

[3] Philips LPC2129 Errata Sheet
http://www.semiconductors.philips.com/acrobat/erratasheets/2129.pdf

[4] ARM7TDMI Technical Reference Manual. Document identity: DDI0029G
http://www.arm.com/pdfs/DDI0029G_7TDMI_R3_trm.pdf

[5] ARM Architecture Reference Manual. Document identity: DDI0100E
Book, Second Edition, edited by David Seal, Addison-Wesley: ISBN 0-201-73719-1
Also available in PDF form on the ARM Technical Publications CD

[6] ARM System Developer's Guide – Designing and Optimizing System Software, by A.N. Sloss, D Symes, C. Wright. Elsevier: ISBN 1-55860-874-5

[7] Embedded System Design on a Shoestring, by Lewin Edwards.
Newnes: ISBN 0750676094.

[8] GNU Manuals
http://www.gnu.org/manual/

[9] GNU ARM tool chain for Cygwin
http://www.gnuarm.com

[10] An Introduction to the GNU Compiler and Linker, by Bill Gatliff
http://www.billgatliff.com

[11] LPC2000 Yahoo Group. A discussion forum dedicated entirely to the Philips LPC2xxx series of microcontrollers.
http://groups.yahoo.com/group/lpc2000/

[12] The Insider's Guide to the Philips ARM7-Based Microcontrollers, by Trevor Martin.
http://www.hitex.co.uk/arm/lpc2000book/index.html

Especially observe document [3]. There exist a number of bugs in the processor that is important to be aware of.

Observe that there can be newer versions of the documents than the ones linked to here. Always check for the latest information / version.

Datasheets for all circuits on the *LPC2129 CAN QuickStart Board* are included on the accompanying CD-ROM.

Copyright 2005 © Embedded Artists AB